

Atelier OpenWrt

Florian Fainelli
Nicolas Thill

/tmp/lab
Vitry sur Seine

3 juillet 2008

OpenWrt
Wireless Freedom

Sommaire I

Le projet OpenWrt

Historique

Whiterussian

Kamikaze

Orientation actuelle

Structure d'OpenWrt

La compilation croisée

Prise en main d'OpenWrt

Aller plus loin

Ajouter un paquet dans OpenWrt

Installer et configurer OpenWrt

Tester OpenWrt

Portage sur une nouvelle architecture

Sommaire II

Portage sur un nouvel quipement partir des sources constructeur
Utiliser le SDK et l'ImageBuilder

Fabriquer un adaptateur RS232 3.3V vers 12V à base de Sipex SP233AC

Historique du projet

- ▶ création fin 2003 par Mike Baker (mbm) et Gerry Rozema (groz)
- ▶ utilisation des sources Linksys WRT54G
- ▶ bas sur buildroot modifi pour supporter IPKG
- ▶ snapshots CVS et uniquement WRT54 et similaires supports

Whiterussian

- ▶ première version stable
- ▶ supporte presque tous les routeurs à base de Broadcom BCM47xx/53xx
- ▶ interface web : Webif
- ▶ se pose le problème de portage vers d'autres architectures
- ▶ difficulté d'intégration de nouveaux paquetages

Kamikaze

- ▶ plus d'architectures supportées : ARM, MIPS, x86, PowerPC, SuperH
- ▶ build system avec plus de modèles et d'abstraction
- ▶ moins de code à écrire pour porter un paquetage dans OpenWrt
- ▶ versions noyaux 2.6 récentes

Orientation actuelle

- ▶ release Kamikaze 8.08 à venir
- ▶ stabilisation des plateformes
- ▶ intégration des patches dans le noyau mainline
- ▶ nouvelle interface web : IUCI
- ▶ environnement pour l'embarqué davantage que firmware pour routeurs

Structure d'OpenWrt

4 répertoires principaux

- ▶ tools
- ▶ toolchain
- ▶ package
- ▶ target

La compilation croisée

- ▶ fichiers de définitions des types de base : kernel headers
- ▶ assembleur et éditeur de liens pour machine cible
- ▶ compilateur pour machine cible
- ▶ librairie C standard
- ▶ ensemble de librairie cross-compilée

Prise en main d'OpenWrt

OpenWrt :

- ▶ téléchargement de tous les outils nécessaires et modification
- ▶ création depuis les sources de la toolchain complète
- ▶ compilation du noyau, root filesystem
- ▶ packaging des programmes
- ▶ création des images avec le bon système de fichiers

Pourquoi ?

- ▶ les PC fonctionnent en 12V, le matériel embarqué en 3.3V
- ▶ nécessité d'adapter les niveaux logiques des tensions
- ▶ 4 broches nécessaires : VCC, GND, TX, RX

Pourquoi le SP233ACP ?

- ▶ pas de nécessité de composants passifs additionnels (capacités, résistances ...)
- ▶ samples gratuits et expédiés rapidement
- ▶ supporte jusqu'à deux UARTs en niveaux logiques 3.3V

Matériel requis

- ▶ fer à souder, étain
- ▶ le SP233ACP
- ▶ un connecteur DB9 femelle
- ▶ un adaptateur USB-série ou port série natif